
UNIVERSITE MOHAMED BOUDIAF DE M'SILA
Faculté des Mathématiques et de l'Informatique
Département de Mathématiques
Présenté pour l'obtention du diplôme de **MASTER**

Domaine : Mathématiques et de L'informatique

Filière : Mathématiques

Option : Analyse Mathématique et Numérique

Présenté par
FELOUSSIA RAYANE

Sujet

**Sur la complexité des algorithmes pour résolution
d'un problème d'optimisation combinatoire**

Soutenu le :27/06/2022

Devant le jury :

Mr.SELT Omar	Prof. Univ de M'sila	Rapporteur
Mr. Bachir Gagui	M.C.A. Univ de M'sila	Président
Mr. Mustapha Dilmi	M.C.B. Univ de M'sila	Examineur

Promotion : 2021/2022

Remerciements

Avant tout, j'adresse mes remerciements en premier lieu, ALLAH tout puissant pour la volonté, le courage et la patience qu'il m'a donné durant toutes ces longues années de formation.

*Je tiens à remercier sincèrement **Mr : SELT Omar**, pour avoir accepté de diriger ce mémoire, pour ses conseils.*

Mes remerciements vont également aux membres du jury qui m'ont fait l'honneur d'examiner ce travail.

Enfin je ne voudrais pas oublier de remercier toute personne qui m'a aidé à réaliser ce travail.

Merci

Dédicace

Je dédie ce modeste travail :

-A mes parents,,

-A mes frères,

-A mes sœurs,

-A toute la famille FELOUSSIA,

-A tous mes amis et toute ma famille de département

de Mathématiques,

-A toutes mes adorables que j'ai connu pendant

toute ma vie ...

FELOUSSIA RAYANE

Table des matières

Remerciements	1
Dédicace	2
Notation	4
1 Optimisation combinatoire	6
1.1 Introduction	6
1.2 Exemples de problème d'optimisation	7
1.2.1 Problème de sac a dos	8
1.3 Fonction maximum	8
1.3.1 solution global	9
1.4 Fonction minimum	9
2 La complexité des algorithmes	11
2.1 la complexité et la théorie de la complexité	11
2.1.1 les algorithmes de la classe P	12
2.1.2 les algorithmes de la classe NP	12
2.1.3 les algorithmes NP Difficiles	13
2.1.4 les algorithmes facile	13
2.1.5 les algorithmes efficace	13
2.1.6 les algorithmes diverge	13
2.1.7 les algorithme converge	14
2.1.8 Résolution d'un problème	14
3 APPLICATION : les algorithmes métaheuristiques	16
3.1 Introduction	16
3.2 Algorithme génétiques	16

3.3	colonie des abeilles	17
3.3.1	les abeilles en nature	17
3.3.2	Recherche de nourriture chez les abeilles	19
3.3.3	Exploration des sources de nourriture	19
3.3.4	Méthode de communication chez les abeilles	19
3.4	Algorithme de colonies d'abeilles	21
3.4.1	Algorithme : L'algorithme de colonies d'abeilles	21
3.5	Généralités sur les fourmis	23
3.6	La métaheuristique « optimisation par les colonies de fourmis »	26
3.6.1	Représentation du problème	27
3.6.2	Construction des solutions	27
3.6.3	l'algorithme de colonie de fourmis	27
3.6.4	Algorithme : L'algorithme de colonies de fourmis	28
3.7	Conclusion générale	29
	Abstract	32

Notation

Nous introduisons les notations et les définitions nécessaires qui sont utilisées par la suite.

p l y	Fonction polynomiale
A B C	Artificiel Bée Colonie
T S P	Traveling Salesman Problème
O P F	d'optimisation de l'écoulement de puissance
F A C T S	Transmission Flexibles en Courant Alternatif
F N C	Forme Normale Canonique
N P	Non déterministe polynomiale temps
P	Polynomiale temps
M T D	Une machine de turing déterministe
T M T D	Le tempe de reconnaissance du problème
A C O	Ante Colonie Optimisation
O C F	Métaheuristique par la les colonies de fourmis

Introduction générale

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulés sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données. Étant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ce mémoire est construit de trois chapitres :

Dans le premier chapitre, On commence par un rappels sur l'optimisation combinatoire, fonction minimum, fonction maximum

Chapitre 2 : Présente La complexité des algorithmes; et on détaille par les sous-titres : classe P, classe NP, NP-Difficiles, facile, les algorithmes efficace, diverge, converge et la résolution d'un problème

Chapitre 3 : C'est une partie purement pratique, elle met en œuvre une technique les algorithmes métaheuristique D'abeilles et les fourmis

Optimisation combinatoire

1.1 Introduction

L'optimisation combinatoire(OC) est un outil indispensable combinant diverses techniques de la mathématique discrète et de l'informatique afin de résoudre des problèmes d'optimisation combinatoire de la vie réelle. Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande. Il s'agit, en général, de maximiser (problème de maximisation) ou de minimiser (problème de minimisation) une fonction objectif sous certaines contraintes .Le but est de trouver une solution optimal dans un temps d'exécution raisonnable. Dans ce chapitre nous présentons la définition de problème d'optimisation combinatoire ;les fonctions min ,max ; les exemples

Définition 1.1. *Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret dit ensemble des solutions réalisables. En général, cet ensemble est fini mais compte un très grand nombre d'éléments, et il est décrit de manière implicite, c'est-à-dire par une liste, relativement courte, de contraintes que doivent satisfaire les solutions réalisables Pour définir la notion de meilleure solution, une fonction, dite fonction objectif, est introduite. Pour chaque solution, elle renvoie un réel et la meilleure solution (ou solution optimale) est celle qui minimise ou maximise la fonction objectif. Clai-
rement, Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret dit ensemble des solutions réalisables. En général, cet ensemble est fini mais compte un très grand nombre d'éléments, et il est décrit de manière implicite,*

c'est-à-dire par une liste, relativement courte, de contraintes que doivent satisfaire les solutions réalisables. Un problème d'optimisation, noté $P(X, f)$, est caractérisé par un ensemble réalisable ou admissible X non-vide et une fonction objectif f qui associe un scalaire dans \mathbb{R} à chaque élément x de l'ensemble X . Les éléments de X sont dits solutions réalisables. Résoudre le problème $P(X, f)$ revient à trouver parmi les solutions réalisables, une qui minimise ou maximise f , c'est-à-dire dans le cas d'un problème de minimisation, trouver une solution $x^* \in X$ telle que $f(x) \geq f(x^*)$ pour tout élément x dans X . Une telle solution est dite optimale et sera notée $x(X, f)$.

Pour définir la notion de meilleure solution, une fonction, dite fonction objectif, est introduite. Pour chaque solution, elle renvoie un réel et la meilleure solution (ou solution optimale) est celle qui minimise ou maximise la fonction objectif. Clairement, un problème d'optimisation combinatoire peut avoir plusieurs solutions optimales. Un problème d'optimisation combinatoire consiste déterminer un plus grand (petit) élément dans un ensemble ni value. En d'autres termes, étant donné une famille F de sous ensembles d'un ensemble ni $E = \{e_1; \dots; e_n\}$ et un système de poids $w = (w(e_1); \dots; w(e_n))$ associé aux éléments de E , un problème d'optimisation consiste à trouver un ensemble $F \in \mathbb{F}$ de poids $w(F) = \sum_{e \in F} w(e)$ maximum (ou minimum), $i : e$; \max ou $\min w(F) F \in \mathbb{F}$. La famille F représente donc les solutions du problème. Elle peut correspondre un ensemble de très grande taille que l'on ne connaît que par des descriptions ou des propriétés théoriques qui ne permettent pas facilement son énumération.

1.2 Exemples de problème d'optimisation

Les problèmes d'optimisation peuvent être des problèmes académiques ou industriels. En fait, les problèmes industriels sont des projections des problèmes académiques sur le plan de la pratique (i.e. la vie réelle). Parmi les problèmes d'optimisation académiques, nous citons le problème de satisfiabilité booléenne *MAX-SAT*, le problème du sac à dos et ses différentes

variantes, le problème du voyageur de commerce et ses différentes variantes, le problème de coloration de graphe, les problèmes d'ordonnancement...etc. Parmi les problèmes industriels nous citons les problèmes de design dans les domaines de l'ingénierie (mécanique, électronique, chimie...) : ailes des avions, moteurs d'automobiles..., les problèmes d'investissements financiers, les problèmes d'ordonnancement en production, l'allocation des usines, la planification de trajectoires de robots mobiles, les problèmes de production agricoles, la gestion de containers, le design de réseaux du transport, les tracés autoroutiers, la distribution de l'eau, le design d'antennes, l'affectation de fréquences...etc. L'objectif de cette section n'est plus de relater les principes des différents problèmes d'optimisation industriels et académiques, mais plutôt d'élucider le principe de quelques problèmes d'optimisation académiques en tant qu'exemples. Une importance particulière est affectée aux problèmes de la famille du problème du sac à dos et aussi au problème du voyageur de commerce car ils sont utilisés dans le test des algorithmes proposés dans le cadre de cette thèse.

1.2.1 Problème de sac a dos

Le problème de sac a dos (en anglais Knapsack problem), modélise une situation analogue au remplissage d'un sac a dos, ne pouvant supporté plus d'un certain poids, avec tout ou une partie d'un ensemble donné d'objet ayant chacun un poids et une valeur. Les objets mis dans le sac a dos doivent maximiser la valeur totale, sans dépasser le poids maximum

1.3 Fonction maximum

Le problème décrit ci-dessus est un problème de minimisation. Comme on a

$$\sup f(x) = - \inf(-f(x)) \quad x \in A$$

un problème de maximisation d'une fonction f (à gauche ci-dessus) est équivalent au problème de minimisation de $-f$ (à droite ci-dessus) . L'équivalence veut dire ici que les solutions sont les mêmes et que les valeurs optimales sont opposées. En particulier, une

méthode pour analyser résoudre un problème de minimisation pourra être utilisée pour analyser résoudre un problème de maximisation.

1.3.1 solution global

Sous certaines conditions, le processus d'optimisation trouve le maximum global. Mais dans certains cas d'optimisation - comme les réseaux de neurones artificiels, le résultat peut être une solution locale. Un maximum local est un point de A tel qu'il existe un voisinage V où pour tout $x \in V$, $f(x) \leq f(\alpha)$. Un minimum local est défini semblablement. Il est en général facile de déterminer numériquement des maxima locaux avec des algorithmes de descentes - comme avec l'Algorithme du gradient. Pour vérifier que la solution trouvée est un maximum global, il est parfois possible de recourir à des connaissances additionnelles sur le

1.4 Fonction minimum

Plus formellement, l'optimisation est l'étude des problèmes qui s'expriment de la manière suivante. Problème d'optimisation étant donné une fonction $f : A \rightarrow \mathbb{R}$ définie sur un ensemble A à valeurs dans l'ensemble \mathbb{R} des nombres réels (éventuellement dans la droite achevée),

$\bar{R} := \mathbb{R} \cup \{-\infty, +\infty\}$ trouver un élément \bar{x} de A tel que $f(\bar{x}) \leq f(x)$ pour tous les x dans A . On dit que l'on cherche à minimiser la fonction sur l'ensemble A . La fonction porte divers noms : fonction-coût ou simplement coût, fonction-objectif ou simplement objectif, critère, etc. L'ensemble A est appelé l'ensemble admissible et les points de A sont appelés les points admissibles du problème (surtout lorsqu'il s'agit d'une partie d'un autre ensemble B et que l'on ne veut pas que k appartienne au complémentaire $\frac{B}{A}$). On dit que le problème est réalisable si A est non vide (l'ensemble admissible étant souvent défini de manière implicite, son caractère non vide n'est pas nécessairement évident, ce qui justifie le besoin de ce concept de réalisabilité). Le point \bar{x} est appelé solution du problème d'optimisation (ou

minimum ou minimiser). On l'appelle aussi parfois une solution globale pour le distinguer des notions locales introduites ci-dessous. On dit qu'il s'agit d'un minimum strict si $\bar{x} \in A$ et $f(\bar{x}) < f(x)$ pour tout $x \in A/\bar{x}$. On peut écrire ce problème de différentes manières :
 $\inf .f(x)$ ou $\inf .f(x|x \in A)$ ou $\inf .f(A)$ ou $\inf .f(x); x \in A$

On note parfois $\arg . \min f(x)|x \in A$ l'ensemble des solutions du problème. L'ensemble $f(A) := f(x)|x \in A$ est une partie de R (ou de \bar{R} si f est valeurs dans \bar{R}) et sa borne inférieure $\inf . f(A)$ est appelée la valeur optimale du problème. Cette valeur optimale est atteinte (c'est-à-dire qu'il existe un $\bar{x} \in A$ tel que $f(\bar{x}) = \inf .f(A)$) si, et seulement si, le problème d'optimisation a une solution. Si $\inf .f(A) > -\infty$ on dit que le problème est borné. On dit que le problème $\inf .f(x)|x \in A$ est convexe si A est une partie convexe d'un espace vectoriel et si f est une fonction convexe sur A

La complexité des algorithmes

2.1 la complexité et la théorie de la complexité

Afin de mesurer la difficulté d'un problème donné et la comparer avec celles d'autres problèmes pour pouvoir dire qu'un tel problème est plus facile à résoudre que l'autre, nous pouvons calculer la complexité algorithmique de chacun d'entre eux. La théorie de la complexité s'intéresse à la complexité d'un problème donné est discutée sur deux cotés : coté temporel (complexité temporelle) et coté spatial (complexité spatiale). La complexité temporelle consiste à évaluer le temps de calcul nécessaire pour résoudre un problème donné. Tandis que la complexité spatiale permet d'estimer les besoins en mémoire (l'espace mémoire requis) pour la résolution d'un problème donné. La complexité d'un problème donné est estimée en fonction du nombre d'instructions permettant d'aboutir à la solution du problème posé. Elle est influencée par la taille du problème en question. En effet, elle exprime un rapport entre la taille du problème, le temps de calcul nécessaire et l'espace mémoire requis. L'évaluation de la difficulté des problèmes via l'étude de la complexité de solutions algorithmiques proposées. Elle associe une fonction de complexité à chaque algorithme résolvant un problème donné et mesure les ressources nécessaires pour la résolution d'un problème posé. En effet, elle classe l'ensemble des problèmes selon deux critères qui sont le temps de calcul nécessaire et l'espace mémoire requis. Bien que la théorie de la complexité se concentre sur des problèmes de décision, elle peut être étendue aux problèmes d'optimisations [13]. Elle classe les problèmes selon leur complexité en deux classes principales : la classe P (Polynomiale temps) et la classe NP (Non déterministe Polynomiale temps).

En outre, elle partage les problèmes de la classe NP en deux sous classes : NP-Complet et NP-Difficile

2.1.1 les algorithmes de la classe P

Les problèmes appartenant à la classe P sont des problèmes pouvant être résolus par une machine de Turing¹ déterministe en temps de calcul polynomial par rapport à la taille de l'instance du problème à traiter. Ils sont souvent faciles à résoudre par des algorithmes efficaces dont le nombre d'instructions nécessaire pour la résolution d'un problème donné est borné par une fonction polynomiale par rapport à la taille du problème. Plus formellement, les problèmes de cette classe sont définis comme suit [8] :

$$P = PL/PLy, \subseteq X^*, \exists MTDPL = PL(MTD), \exists Ply, \forall w \in pL = n, T_{MTD}(w) < Ply(n)$$

Où Ply est une fonction polynomiale, MTD une machine de Turing déterministe, PL un problème et T_{MTD} le temps de reconnaissance du problème. Le calcul du plus grand diviseur commun est un exemple d'un problème appartenant à la classe P.

2.1.2 les algorithmes de la classe NP

La classe des problèmes NP regroupe l'ensemble de problèmes qui peuvent être résolus avec une machine de Turing non-déterministe² et admettent un algorithme polynomial pour tester la validité d'une solution du problème traité (i.e. il est possible de vérifier que la solution proposée est correcte en un temps polynomial par rapport à la taille du problème). Le problème du sac à dos, le problème du voyageur de commerce, les problèmes d'ordonnancement, le problème de coloration de graphes sont des exemples de problèmes appartenant à la classe NP.

1. Une machine abstraite représentant un modèle abstrait du fonctionnement de l'ordinateur et de sa mémoire. Elle a été proposée par le mathématicien Alan Mathison Turing en 1936 en vue de donner une définition précise au concept d'algorithme.

2. Le terme non- déterministe désigne un pouvoir qu'on incorpore à un algorithme pour qu'il puisse aboutir à la bonne solution.

2.1.3 les algorithmes NP Difficiles

la classe de problèmes NP-Difficiles englobe les problèmes de décision et les problèmes d'optimisation. Les problèmes NP-Difficiles sont aussi difficiles que les problèmes NP- Complets. Si un problème de décision associé à un problème d'optimisation P est NP- Complet alors P est un NP-Difficile [8]. Par conséquent, afin de prouver qu'un problème d'optimisation est NP-Difficile, il suffit de montrer que le problème de décision associé à P est NP-Complet [3]. Il est à noter que jusqu'à maintenant, aucun algorithme polynomial n'est connu pour résoudre ce type de problèmes (i.e. NP- Difficiles).

2.1.4 les algorithmes facile

Un problème P est un problème facile à résoudre. Il possède une solution qui peut être trouvée dans un temps polynomial avec la taille de l'instance traitée.

2.1.5 les algorithmes efficace

Efficacité d'un algorithme par rapport à un autre « On considère généralement qu'un algorithme est plus efficace qu'un autre si son temps d'exécution du cas le plus défavorable a un ordre de grandeur inférieur. » C'est pour ces raisons que la notation $O(10)$ (Notation de Landau) est souvent utilisée pour la mesure de la complexité car elle décrit une borne supérieure du cas le plus défavorable.

2.1.6 les algorithmes diverge

L'opérateur divergence est un outil d'analyse vectorielle qui mesure, pour faire simple, si un champ vectoriel « rentre » ou « sort » d'une zone de l'espace, comme ce que l'on peut observer sur un diagramme de lignes de champ. Il donne donc une information très liée aux sources qui créent le champ. Les équations de continuité permettent de comprendre intuitivement cette notion, la divergence va en effet mesurer localement les variations de densité de flux, on retrouve cette grandeur, macroscopique cette fois, dans les valeurs de

diffusion de particule ou de chaleur par exemple. Ainsi, en un point, si la divergence est nulle, alors la densité ne varie pas et si elle est positive en ce point, alors il y a diffusion

2.1.7 les algorithmes convergents

L'algorithme converge, alors la limite est un point stationnaire pour la somme des carrés S . Toutefois, la convergence n'est pas garantie, pas plus qu'une convergence locale contrairement à la méthode de Newton. La vitesse de convergence de l'algorithme de Gauss-Newton peut approcher la vitesse quadratique. L'algorithme peut converger lentement voire ne pas converger du tout si le point de départ de l'algorithme est trop loin du minimum. L'algorithme peut donc échouer à converger. Par exemple, le problème avec $m = 2$ équations et $n = 1$ variable, donné par

$$r_1(\beta) = \beta + 1$$

$$r_2(\beta) = \lambda(\beta^2) + \beta - 1.$$

L'optimum se situe en $\beta = 0$. Si $\lambda = 0$ alors le problème est en fait linéaire et la méthode trouve la solution en une seule itération. Si $|\lambda| < 1$, alors la méthode converge linéairement et les erreurs décroissent avec un facteur $|\lambda|$ à chaque itération. Cependant, si $|\lambda| > 1$, alors la méthode ne converge même pas localement.

2.1.8 Résolution d'un problème

Les méthodes de résolution des problèmes d'optimisation sont de deux types :

- Les méthodes exactes (déterministes) : elles fournissent une solution optimale au prix d'un temps de résolution qui risque d'être exponentiel en fonction de la taille des données du problème.
- Les méthodes approchées : pour un problème d'optimisation dit « difficile » aucune méthode exacte n'est capable de le résoudre exactement en un temps raisonnable. Dans ce cas on fait appel à ses méthodes permettant une optimisation approchée. Ce type de méthodes retourne une solution contenue dans un certain intervalle autour de la solution

optimum avec un temps de calcul acceptable. Elles représentent un compromis entre la qualité de la solution trouvée et le temps de calcul nécessaire

APPLICATION : les algorithmes métaheuristiques

3.1 Introduction

Les métaheuristiques sont un ensemble d'algorithmes d'optimisation visant à résoudre les problèmes d'optimisation difficiles. Elles sont souvent inspirées par des systèmes naturels, qu'ils soient pris en physique (cas du recuit simulé), en biologie de l'évolution (cas des algorithmes Génétiques) ou encore en éthologie (cas des algorithmes de colonies de fourmis ou de l'optimisation par essaims particulaires).[10] Ces techniques métaheuristiques peuvent êtres classées en deux groupes : les méthodes à population de solutions connues sous le nom d'algorithmes évolutionnaires comme les algorithmes génétiques...etc., ainsi que les méthodes à solution unique comme le recuit simulé. Les méthodes métaheuristiques ont prouvé leurs efficacités dans le domaine de l'optimisation mono-objectif. Actuellement les recherches qui utilisent ces algorithmes sont développées pour la résolution des problèmes d'optimisation multi objectif, en tenant compte de plusieurs contraintes et de nouvelles configurations des réseaux électriques surtout à l'associations de sources des énergies renouvelables où la résolution de ce système complexe est un défi

3.2 Algorithme génétiques

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes. Leur but est d'obtenir une solution approchée à un problème d'optimisation [7]. La solution optimale est cherchée à partir d'une population de solutions en utilisant des processus

aléatoires. La recherche de la meilleure solution est effectuée en créant une nouvelle génération de solutions par application successive, à la population courante, de trois opérateurs : la sélection, le croisement, et la mutation. Ces opérations sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint

3.3 colonie des abeilles

L'optimisation par colonie d'abeilles est une famille très récente des métaheuristiques. Son principe est basé sur le comportement des abeilles réelle dans la vie. Cette approche de résolution fait l'objet de notre étude. Dans ce contexte, nous allons donner la structure d'une colonie d'abeille, puis une description du comportement des abeilles lors de la recherche de nourriture, enfin nous allons présenter quelques algorithmes inspirés du ce comportement.

3.3.1 les abeilles en nature

Les abeilles possèdent des propriétés assez différentes de celles des autres espèces d'insectes. Elles vivent en colonies, en construisant leurs nids dans des troncs d'arbres ou d'autres espaces clos similaires [8]. Généralement, une colonie d'abeilles contient une femelle reproductrice appelée reine, quelques centaines de mâles connus sous le nom de faux-bourdon, et de 10.000 à 80.000 femelles stériles qui s'appellent les ouvrières. Après accouplement avec plusieurs faux-bourdon, la reine reproduit beaucoup de jeunes abeilles appelées les couvées.

La Reine : Dans une colonie d'abeilles, il y a une seule reine qui est la femelle reproductrice avec l'espérance de vie entre 3 et 5 ans. Son rôle principal est la reproduction, elle s'accouple avec (7-20) faux-bourdon dans une opération de reproduction appelée le vol nuptial (mating flight). Elle conserve les spermés dans sa spermathèque et puis pond jusqu'à 2000 œufs par jour. Les œufs fertilisés vont bien devenir femelles (ouvrières) et les œufs non-fertilisés deviennent mâles (bourdons).

Le mâle (faux-bourdon) : Au sein de la famille des Apidés, se trouvent plusieurs

genres, et notamment les bourdons, qu'il ne faut pas confondre avec les faux-bourdons, les mâles de l'abeille domestique. Les faux-bourdons représentent les mâles, variant entre 300 et 3000 dans une ruche. Selon la taille de l'alvéole, dans une grande cellule hexagonale, la reine dépose un œuf sans que son réceptacle séminal laisse sortir de spermatozoïde. L'œuf non fécondé donnera ainsi naissance à un faux-bourdon. Ce dernier a une espérance de vie de 90 jours. Après l'accouplement, qui a lieu en vol, le mâle meurt rapidement [1].

Ouvrières : Les ouvrières sont les abeilles femelles mais elles ne sont pas reproductrices, elles vivent de 4 à 9 mois dans une saison froide et leur nombre arrive jusqu'à 30.000 dans une colonie (ruche). Cependant, en été, leur durée de vie est de 6 semaines et leur nombre atteint jusqu'à 80.000. L'ouvrière est responsable de la défense de la ruche utilisant sa piqûre barbelée. En conséquence, elle meurt après avoir piquée. On peut énumérer les activités des ouvrières par le critère des jours de sa vie comme suit : nettoyage de cellules (jour 1-2), soigner les abeilles (jour 3-11), production de cire (jour 12-17), surveiller les autres abeilles (jour 18-21), et recherche de nourriture (jour 22-42). L'ouvrière assure les activités habituelles de la colonie d'abeilles telles que l'emballage de pollen, éventer les abeilles, porter l'eau, déplacement des œufs, s'occuper de la reine, nourrir les bourdons, et construction du nid d'abeilles [9].

Couvées : Les jeunes abeilles sont nommées les couvées. Elles sont nées après que la reine pond un œuf fécondé par un spermatozoïde libéré par la spermatique dans des cellules spéciales du nid d'abeilles appelées trames de couvée (brood frames). L'œuf fécondé donne naissance à une abeille ou à une reine en fonction des soins et de la nourriture apportés aux larves. Peu de larves femelles sont choisies pour être des futures reines. Les œufs non fertilisés donnent naissance aux couvées. Les jeunes larves tournent par le cocon, couvrant la cellule par les ouvrières les plus âgées. C'est l'étape de chrysalides. Puis, elles atteignent l'étape de développement dans laquelle elles reçoivent le nectar et le pollen des abeilles chercheurs jusqu'à ce qu'elles quittent la ruche et passent leurs vies comme chercheurs de

nourriture [9]

3.3.2 Recherche de nourriture chez les abeilles

Les scientifiques ont effectué beaucoup de recherches pour déterminer comment l'ordre est maintenu dans la ruche où vivent des dizaines de milliers d'abeilles. Un grand nombre d'études académiques ont été aussi effectuées à cette fin. Un éminent expert et professeur à l'Université de Munich, le zoologiste autrichien Karl Von Frisch, a consacré un livre de 350 pages à la communication des abeilles, "The dance language and orientation of bees" (Le langage de la danse et l'orientation des abeilles) [8]

3.3.3 Exploration des sources de nourriture

L'abeille qui cherche à manger, appelée « éclaireuse », se trouve devant une tâche formidable. Elle quitte la ruche, cherche dans de vastes zones et vole de longues distances, elle doit donc reconnaître les bonnes sources de nourriture et avant que les abeilles butineuses retournent à la ruche, elle doit enregistrer sa position par rapport à la ruche et dépose une odeur spéciale sur leur source de nourriture. Une fois qu'une source de nourriture a été localisée et marquée, elle doit retourner à la ruche pour informer les autres ouvrières de sa découverte. Pour cela, elle doit utiliser des informations sur la direction de la ruche même si, pour arriver où elle est, elle fait tout un circuit. Après, elle doit communiquer aux autres ouvrières la direction et la distance de la source de nourriture par rapport à la ruche, et elle doit donner certaines indications sur sa qualité. Ceci doit être fait de la façon la plus économique possible. Elle doit donc attirer l'attention des autres ouvrières, qui peuvent déjà être occupées à autre choses ou avoir reçu des messages

3.3.4 Méthode de communication chez les abeilles

Karl Von Frisch a construit une ruche avec une paroi en verre par laquelle il pouvait observer le comportement des abeilles à l'intérieur. Il remarqua que dans le retour des

abeilles employeuse à la ruche, elles exécutent la danse. Si une seule abeille l'exécute, ce n'est pas toute la ruche qui passe à l'action. Premièrement, un groupe d'éclaireuses quitte la ruche. Si, à son retour, ce groupe exécute aussi la danse, alors un groupe plus important d'abeilles se dirigera vers l'objectif. Meilleure est la source de nourriture qu'elles trouvent, plus longtemps dure la danse et plus grand sera le nombre d'abeilles qui les suivent. De cette façon, l'attention des butineuses est toujours fixée sur la source la plus productive

Si la source de nourriture qu'elles trouvent est très riche, la danse que les abeilles exécutent est très enthousiaste. Si la source est à proximité, elles décrivent son emplacement en exécutant la "danse en rond" illustrée à droite. Pour les sources de nourriture plus éloignées, elles exécutent la danse en huit illustrée à gauche, avec des mouvements frétilants [8]. Dans la danse frétilante, l'abeille tourne alternativement d'un côté et de l'autre effectuant un demi tour circulaire qui la ramène au point de départ. Les ouvrières suivent la danseuse au cours de ses déplacements, et la touchent des antennes. Cette danse est très riche en information, en particulier, elle donne aux abeilles observatrices deux indications essentielles : la quantité de nourriture, la distance à parcourir et la direction à suivre [2]. Concernant la distance, plusieurs paramètres sont utilisés simultanément, le plus important est le tempo de la danse, c'est-à-dire le nombre de tours complets effectués par la danseuse par unité de temps. Les mouvements de l'abeille sont d'autant plus durables que la source est riche (quantité de nourriture). En ce qui concerne la direction, elle est donnée par rapport à l'azimut solaire (en plan horizontal), la position angulaire de la source est indiquée par l'angle que fait la partie frétilante avec la verticale du rayon comme l'indique [8]. Dans le cas où la source de nourriture trouvée serait improductive, les abeilles dansent tout de même, mais elles le font à contrecœur et pendant une plus brève durée. Cela se reflète aussi sur les autres abeilles dans la ruche, les abeilles qui se rassemblent autour de la danseuse se dispersent rapidement et une nouvelle équipe quitte la ruche à la recherche de nourriture [8].

1- Si la source de nourriture se trouve exactement dans la direction du soleil, ou dans la direction opposée, les frétillements dans la danse seront verticaux sur le rayon.

2- Si la source de nourriture est 80 degrés à gauche du soleil, cela est indiqué en faisant la trajectoire du frétillement de la danse à un angle correspondant de 80 degrés à gauche de la verticale.

3- Si l'abeille suit une direction vers le haut durant son frétillement, elle signale que la source de nourriture se trouve dans la direction vers le soleil. Et si elle se dirige droit en bas, cela signifie que la source se trouve dans la direction exactement opposée de celle du soleil

3.4 Algorithme de colonies d'abeilles

L'optimisation par colonies d'abeilles artificielles (ABC : Artificielle Bée Colonie) est une nouvelle méta-heuristique qui a enrichi le nombre des méthodes d'optimisation basées sur l'intelligence par essaim. Elle a été proposée en 2005 par Karaboga [14].

3.4.1 Algorithme : L'algorithme de colonies d'abeilles

Début
Initialiser une population de N solution ;
Évaluer les N solutions ;
Cycle=1 ;
Tant que cycle \leq MCN faire Construire une nouvelle solution v_i pour chaque ouvrière i en utilisant l'équation , puis les évaluer ;
Sélectionner les ouvrières ;
Calculer les valeurs de probabilités P_i pour les solutions x_i en utilisant l'équation ;
Construire la nouvelle solution de chaque abeille spectatrice à partir de la solution x_i sélectionnée en fonction de la probabilité p_i ;
Évaluer les nouvelles solutions ; Sélectionner les spectatrices ;
Déterminer les solutions à abandonner par les scoutes si elles existent et les remplacer par des solutions aléatoires ;
Enregistrer la meilleure solution trouvée. Cycle= Cycle+1 ;
Fin Tant que
Retourner la meilleure solution ;
Fin

L'algorithme ABC s'inspire du modèle naturel du comportement des abeilles mellifères lors de la recherche de leur nourriture. Le processus de recherche de nourriture chez les abeilles est fondé sur un mécanisme de déplacement très efficace. Il leur permet d'attirer l'attention d'autres abeilles de la colonie aux sources alimentaires trouvées dans le but de collecter des ressources diverses. En fait, les abeilles utilisent un ensemble de danses frétilantes comme moyen de communication entre elles. Ces danses permettent aux abeilles de partager des informations sur la direction, la distance et la quantité du nectar avec ses congénères. La collaboration et la connaissance collective des abeilles de la même colonie sont basées sur l'échange d'information sur la quantité du nectar dans la source de nourriture trouvée par les différents membres. Des études sur le comportement de danses frétilantes des abeilles ont montré [14] La direction des abeilles indique la direction de la source de nourriture par rapport au soleil L'intensité de la danse indique la distance de la source de [8] nourriture.

La durée de la danse indique la quantité du nectar dans la source de nourriture trouvée.

Dans un algorithme d'optimisation par colonies d'abeilles, une source de nectar correspond à une solution possible au problème à traiter. La colonie d'abeilles artificielle est composée de trois types d'abeilles : les ouvrières, les spectatrices et les scoutes.

- L'ouvrière exploite la source de nourriture trouvée. Elle se base sur sa mémoire et essaye d'apporter des modifications à sa position (solution) actuelle pour découvrir une nouvelle position (i.e. source de nourriture).

- L'abeille spectatrice attend le retour des ouvrières au champ de danse pour observer leurs danses et recueillir des informations sur les sources de nectar qu'elles ont trouvées.

- L'abeille scoute exploite l'espace de recherche en lançant une recherche aléatoire d'une nouvelle source de nourriture. Une abeille ouvrière est assignée à chaque source de nourriture. La taille de la population de la colonie est égale au nombre des abeilles ouvrières et la quantité du nectar dans une source de nourriture correspond à la qualité (fitness) de

la solution proposée. L'algorithme représente un pseudo code de l'algorithme de colonies d'abeilles.

3.5 Généralités sur les fourmis

Avec les abeilles et les guêpes, les fourmis font partie de la famille des formicidés, incluse dans l'ordre Hyménoptères, c'est à dire des insectes dont les deux paires d'ailes sont membraneuses et fines. Les fourmis sont réparties en 11 sous-familles, et approximativement 10 000 espèces. La fourmi peut mesurer de quelques millimètres à plusieurs centimètres de long. Son corps qui se compose de 3 parties principales : la tête, le thorax et l'abdomen. La tête est ornée d'une paire d'antennes qui abritent plusieurs organes des sens : le toucher, l'odorat et le goût. On trouve également à l'extrémité de la tête, la bouche qui sert à la fois à l'alimentation et très souvent d'outil de saisie. Les yeux, au nombre de 2, sont composés et formés d'une multitude de facettes. Quelques fourmis sont complètement aveugles, sans aucun organe visuel. Le thorax, ou corselet, porte 3 paires de pattes, très longues et terminées par 2 griffes. Les organes tactiles sont bien développés et sont constitués par des poils répartis sur tout le corps. Leur densité est plus forte sur les antennes et aux extrémités des pattes. L'abdomen est relié au thorax par un étroit pédoncule ou pétiole. L'abdomen renferme le jabot social, l'estomac, l'intestin et les conduits respiratoires, et chez les reines et les mâles, l'appareil reproducteur. La fourmi a un estomac un peu spéciale, le "jabot social". Elle y garde en réserve de la nourriture, qui peut être régurgitée et transmise aux autres fourmis. La carapace est composée d'une substance, résistante et imperméable que l'on nomme chitine. Elle renferme différents pigments, bruns, rouges, ou noirs qui sont responsables de la couleur de l'animal. Selon les espèces, les fourmis vivent dans le sol, sous un caillou, dans le bois mort ou dans les arbres. Certaines fourmis arboricoles construisent leurs nids dans les arbres, en cousant les feuilles entre elles ou en élaborant une architecture faite de grains de sable, de salive et de déjections. D'autres, comme les célèbres fourmis "lé-

gionnaires” d’Afrique, n’ont pas de nid est déplacent par centaines de milliers d’individus sur de vastes territoires, à la manière des nomades. La fourmi est un insecte exclusivement social vivant en société dans des habitations collectives : la fourmilière. La fourmilière classique est constituée par un ensemble de cellules réunies entre elles par un réseau complexe de galeries qui peut être très important. Dans ces cellules, les fourmis déposent les œufs, les larves et les nymphes. La fourmilière et ses environs constituent le centre de la vie communautaire. A l’intérieur de la fourmilière les tâches sont divisées entre les fourmis selon la spécialité de chacune d’elles. Les activités des communautés de fourmis sont caractérisées par un certain degré de division du travail souligné par une différenciation fonctionnelle et anatomique des individus. Une fourmilière peut abriter de 50.000 à plus de 1.000.000 individus bien différenciés tant au niveau physique qu’au niveau des comportements et des tâches à accomplir. On les sélectionne en castes :

Les reines : Dans une fourmilière on trouve une ou plusieurs reines. Les reines sont nettement plus grosses que les autres fourmis et peuvent vivre jusqu’à dix ou quinze ans. Leur rôle se résume essentiellement à pondre des œufs et sont donc les fondatrices de nouvelles colonies..

Les ouvrières : Elles forment la majorité des habitants de la cité et se chargent de la défense et de l’entretien de la colonie, qui comprend la construction des galeries, les soins apportés aux jeunes, la quête de la nourriture, etc...

Les soldats : Ils sont plus massifs que les ouvrières, et possèdent souvent de grosses mandibules. Leur rôle est de défendre la fourmilière, et de transporter des charges lourdes. Mais certains, comme chez les fourmis *Campus*, macule, participent aux soins des larves, et donnent à manger aux fourmis qui le demandent. Chez cette espèce, qui est la plus grosse fourmi vivant à La Réunion, on peut distinguer des formes intermédiaires entre la petite ouvrière grêle et le puissant soldat.

Les jeunes sexués : ce sont des fourmis femelles et mâles. Ils sont facilement recon-

naissables par leur plus grande taille, par la présence de deux paires d'ailes membraneuses sur le thorax et par 3 ocelles disposés en triangle sur le dessus de la tête. Les femelles sont plus grosses que les mâles. Ce sont les futurs rois et reines qui iront fonder de nouvelles colonies. Le couvain Il est constitué par les œufs, les larves et les nymphes. Au bout de quelques jours les œufs donnent naissance à des larves qui, bien nourries par les ouvrières pendant 15 jours à 3 semaines, se transforment en nymphes. Pendant la nymphose, la larve ne se nourrit plus Son corps tout entier subit de profondes mutations internes et externes, qui vont faire d'elle, petit a petit, une fourmi. Les communications interindividuelles entre fourmis sont de plusieurs types et varient d'une espèce a l'autre. Les principaux moyens de communication sont : La communication tactile : lorsque deux fourmis se rencontrent, elles procèdent a quelques attouchements d'antennes ou de pattes, et se reconnaissent aussitôt comme membres de la même fourmilière; La communication sonore : les fourmis peuvent également utiliser des stimuli vibratoire comme moyen de communication. Elles frappent les parois de la fourmilière avec leur abdomen pour prévenir les autres. Elles tapotent aussi leurs antennes pour se parler. Ce signal est utilise en fonction de l'espace comme signal de détresse en cas de danger, comme signal de qualité de l'alimentation pour le recrutement pour une source de nourriture, comme signal de demande d'aide en cas ou la nourriture trouve est de grande taille.. La communication visuelle : développée chez certaines espèces, permet aux fourmis de retrouver leur chemin, guidées par des marqueurs visuels. La communication chimique : les fourmis sont équipées de glandes produisant des phéromones, substances chimiques volatiles et odorantes qu'elles peuvent sentir par leurs antennes. Ce signal chimique porte l'information a la fois sur l'espace, la société mais aussi la caste et le stade de développement auxquelles appartiennent les fourmis rencontrer. C'est en sécrétant cette substance qu'une fourmi éclairieuse marque le chemin qu'elle a utilise et revient avertir ses pairs de la présence de nourriture ou d'un danger, ainsi que de sa localisation. C'est le moyen de communication le plus efficace et plus utilisé par toutes les espèces de fourmis

La totalité des espèces de fourmis vit en effet en communautés 49 plus ou moins importantes, qui sont qualifiées d'eu-sociales par [12], ce qui signifie qu'elles sont caractérisées par la présence simultanée et constante des trois critères suivants :

1. La coopération des membres de la société pour élever les jeunes et le couvain ;
2. L'existence d'une division des rôles, d'abord au niveau de la fonction reproductrice, ensuite au niveau des tâches à accomplir
3. Le chevauchement d'au moins deux générations d'âge différent capables de contribuer aux travaux de la colonie.

3.6 La métaheuristique « optimisation par les colonies de fourmis »

Les algorithmes qu'on vient de décrire ont été regroupés sous le terme général de « métaheuristique par les colonies de fourmis (O C F) » ou « Ante Colonie Optimisation (A C O) » [10]. Dans O C F, le voisinage du nid des fourmis est associé à l'espace des solutions, telle que chaque solution s'apparente à une source de nourriture dont la qualité est fournie par une fonction objectif. Chaque fourmi est assimilée à un processus stochastique de construction de solutions itératif. La construction est biaisée par une phéromone artificielle et une information heuristique propre au problème à résoudre. Les traces de phéromones sont typiquement la mémoire des fourmis sur les éléments de la solution, et sont régulièrement mise à jour par les processus de construction (en fonction de la qualité de la solution) et par un mécanisme simulant l'évaporation de la phéromone. D'une manière générale, l'utilisation de O C F nécessite de choisir une représentation formelle du problème d'optimisation à traiter et de définir le processus de construction de solutions par les fourmis en utilisant cette représentation.

3.6.1 Représentation du problème

Un problème d'optimisation $O = \{S, f, \Omega\}$ est représenté par un ensemble de solutions S , une fonction objectif f assignant une qualité à une solution et un ensemble de contraintes Ω . L'objectif est de trouver une solution optimale $f_{\min}(s), \forall s \in S$ qui satisfait les contraintes Ω .

Suivant cette représentation, les fourmis construisent les solutions possibles en se déplaçant sur un graphe $G = (C, L)$ associé au problème d'optimisation, tel que les nœuds sont les composantes C du problème qui forment les solutions et les arcs sont les transitions possibles L qui connectent les composantes de C , telle que à chaque transition on associe une fonction de coût. Les contraintes du problème sont implémentées directement dans les règles de déplacement des fourmis. Par conséquent, la solution optimale représente le chemin de Coût minimum.

3.6.2 Construction des solutions

Les fourmis artificielles construisent les solutions en ajoutant selon un processus stochastique et itératif les séquences de composants à partir de C . Chaque fourmi construit une solution possible (même de mauvaise qualité) d'une manière stochastique en se déplaçant dans le graphe $G=(C,L)$. La solution optimale émergera de la coopération entre les fourmis de la colonie.

3.6.3 l'algorithme de colonie de fourmis

L'algorithme de colonies de fourmis est un des algorithmes basés sur l'intelligence par essaim. Il a été introduit au début des années 90 par le trinôme Colonie. L'idée de base du trinôme imite le comportement collectif des fourmis lors de leur déplacement entre la fourmilière et la source de nourriture. L'objectif du comportement collectif des fourmis est de collecter la nourriture sans perdre le chemin menant à leur nid. Les fourmis sont des insectes qui œuvrent pour le bien du groupe. Leurs capacités physiques et cognitives

limitées n'ont jamais construit un obstacle pour elles. En effet, elles peuvent défier leurs capacités individuelles limitées et réaliser des tâches très complexes (construire des nids, rechercher la nourriture, élever les larves...) par coopération en regroupant leurs capacités disponibles et leurs expériences collectives. Dans l'objectif de rechercher la nourriture en parcourant le plus court chemin, les fourmis se communiquent indirectement entre elles en provoquant des changements dans leur environnement. Au début de la recherche, les fourmis se propagent aléatoirement en prenant des chemins de différentes tailles (court, long,...) dont elles déposent sur le sol une matière odorante appelée «phéromone» d'intensités égales. Afin d'attirer l'attention de leurs congénères en retournant au nid, les fourmis déposent des phéromones un peu différents contenant un message concernant la qualité du site visité. Les fourmis ont tendance de suivre le chemin de plus forte intensité de phéromones. Plus le chemin est court, plus la quantité de phéromones y est déposée est élevée. Et plus l'intensité de phéromones est grande, plus le nombre de fourmis utilisant ce passage augmente. Par conséquent, le chemin le plus long sera abandonné car l'intensité de phéromones y compris est petite et s'évapore rapidement. L'auto-organisation des fourmis basée sur l'utilisation des marqueurs chimiques (i.e. les phéromones) a construit le secret du succès de ces insectes dans la résolution de leurs problèmes. La fourmi informatique représente une solution au problème traité. Les phéromones informatiques sont des valeurs associées à des solutions trouvées. Ces valeurs dépendent des qualités des solutions. En fait, chaque fourmi (i.e. solution) dépose une certaine quantité de phéromones qui dépend de sa qualité. L'algorithme de colonies de fourmis a été proposé pour la première fois pour résoudre le problème du voyageur de commerce, il se base sur trois phases essentielles :

3.6.4 Algorithme : L'algorithme de colonies de fourmis

- La construction du trajet de chaque fourmi.
- La distribution de phéromones sur le trajet de chaque fourmi.
- Évaporation des pistes de phéromones.

Début

Initialiser une population de m fourmis; Évaluer les m fourmis; Tant que la condition d'arrêt n'est pas satisfaite faire

Pour $i=1$ à m faire

Construire le trajet de la fourmi i ;

Déposer des phéromones sur le trajet de la fourmi i ;

Fin pour

Évaluer les m fourmis;

Évaporer les pistes de phéromones;

Fin Tant que

Retourner la ou les meilleures solutions;

Fin

L'algorithme représente le schéma général de l'algorithme de colonies de fourmis pour le problème du voyageur de commerce (T S P : Traveling Salesman Problème). Il est à noter qu'à part le problème du voyageur de commerce, l'algorithme de colonies de fourmis a été appliqué avec succès sur d'autres problèmes d'optimisation comme : les problèmes des tournées de véhicules, le problème d'affectation quadratique...etc.

3.7 Conclusion générale

Dans ce mémoire, nous avons effectué une étude la complexité d'algorithmes pour résolution d'un problème d'optimisation combinatoire. En particulier, nous avons présenté les algorithmes de classe P, NP, les algorithmes génétiques est pour résolution d'un problème d'optimisation on va suivre deux types de méthodes sont : Les méthodes exactes : elle fournissent une solution optimale au prix d'un temps de résolution qui risque d'être exponentiel en fonction de la taille des données du problème. Les méthodes approchées : Ce type de méthodes retourne une solution contenue dans un certain intervalle autour de la solution optimum avec un temps de calcul acceptable. Elles représentent un compromis entre la qualité de la solution trouvée et le temps de calcul nécessaire. Les algorithmes approchés efficaces pour les problèmes d'optimisation combinatoire sans garantie optimale ; le but est de trouver des meilleures solutions, ou bien ajouter des contraintes ou des systèmes complexes pour filtrer une bonne population initiale ce qui va aider de trouver une solution optimale.

Bibliographie

- [1] Glover, F. (1986). *Future paths for integer programming and links to artificial intelligence*, *Comput. Oper. Res.*, 13, 533-549.
- [2] Glover, F., and S. Hanafi, S. (2002). *Tabu Search and Finite Convergence, Special Issue on Foundations of heuristics in Combinatorial Optimization*. *Discrete Appl. Math*, 119, 3- 36.
- [3] Hansen, P. (2006). *The steepest ascent mildest descent heuristic for combinatorial programming*, *Proceedings of the Congress on Numerical Methods, Capri, Italy*.
- [4] Haouari, M., and Ladhari, T. (2003) *A branch-and-bound-based local search method for the flow shop problem*, *J. Oper. Res. Soc.*, 54, 1076-1084. DOI : 10.1057/palgrave.jors.2601612
- [5] Lee, C. Y. (1996). *Machine scheduling with an availability constraints*, *J. Global Optim.*, 9, 395-416.
- [6] Lee, C. Y. (1997). *Minimising the makespan in two machines flow shop scheduling problem with availability constraints*, *Oper. Res. Lett.*, 20, 129-139.
- [7] M'Hallah, R., and Bulfin, R. L. (2005). *Minimizing the weighted number of tardy jobs of parallel processors*, *Eur. J. Oper. Res.*, 160, 471-4847.
- [8] Sakarovitch, M. (1984). *Optimisation combinatoire : Programmation discrete*, Hermann, France.
- [9] Schmidt, G. (2000). *Scheduling with limited machine availability*, *European J. Oper*, 121, 1-15.

- [10] Selt, O.,and Zitouni, R. (2014).*A comparative study of heuristic and metaheuristic for three identical parallel machines, Cjpas, 3147-3153.*
- [11] Smith, W. E. (1956).*Various optimizes for single-stage production, NavaRes.Logist, 3, 59- 66.*
- [12] Liang, Y.-C., Hsiao, Y.-M.,and Tien, C.-Y. (2013).*Metaheuristics for drilling operation scheduling in Taiwan PCB industries, International Journal of Production Economics, 141(1), 189-198. DOI : 10.1016/j.ijpe.2012.04.014*
- [13] Zribi, N., Kacem, I., El-Kamel, A., and Borne, P. (2005).*Minimisation de la somme des retards dans un job shop flexible, Revue e-STA (SEE), 6(6), 21-25.*
- [14] Zitouni, R.,and Selt, O. (2016). *Metaheuristics to solve tasks scheduling problem in parallel identical machines with unavailability periods, RAIRORes, 50(1), 90.*

Abstract

In this work we present some problems of combinatorial optimization, and we present a definition of complexity algorithm for resolution same problems NP-difficult.

Key words : combinatorial optimization ; algorithm ; NP-difficult.

Résumé

Dans ce travail nous avons étudié Quelques problèmes d'optimisation combinatoire et nous avons présenter la notion general de la complexite des algorithmes pour résoudre quelque probleme NP-Difficile.

Mots clés : optimisation combinatoire ; Algorithmique ; NP-Difficile.

ملخص

الهدف في هذه المذكرة قمنا بالتطرق الى بعض مسائل التحليل التوافقي، كما عرفنا بعض خواص الخوارزميات الحدودية وتقاربها الى حل مسألة توفيقية واخيرا اعطينا بعض الامثلة على خوارزميات ميتاوريستيك وفاعليتها في حل مسائل ذات حجم القياس كلمات مفتاحية : التحليل التوافقي، الخوارزمية، غير كثيرة حدود الصعبة.